

Firemní archiv souborů

Company Files Archive

Souhlasím se zveřejněním této bakalářské práce dle požadavků čl. 26, odst. 9 *Studijního a zkušebního řádu pro studium v bakalářských programech VŠB-TU Ostrava*.

V Ostravě 6. května 2011

.....

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

V Ostravě 6. května 2011

.....

Rád bych na tomto místě poděkoval svému vedoucímu bakalářské práce, panu Markovi Běhálkovi, za náměty pro zdokonalení práce, pomoc a odborné vedení při tvorbě bakalářské práce.

Abstrakt

Cílem této bakalářské práce je navrhnout a vytvořit systém zálohování a archivace dat pro společnost Vodovody a kanalizace Přerov, a.s. Práce se zabývá vývojem aplikace od samotného zadání, přes jeho návrh až po konečnou implementaci, nasazení a testování v provozu. Systém se bude skládat ze dvou základních částí, a to z klienta a serveru. Klient bude umožňovat zálohování souborů na lokální úložiště, jejich kompresi, plánovat a automaticky spouštět zálohovací proces, synchronizaci a připojení na centrální datový server pomocí sítě LAN a obnovení dat z již vytvořených záloh. Server bude umožňovat centrální řízené ukládání záloh, správu jednotlivých verzí uložených dat, jejich deduplikaci na souborové úrovni, paralelní připojení jednotlivých klientů a jejich správu, obnovu konkrétních dat do libovolného umístění a administrační rutiny pro kontrolu databáze.

Klíčová slova: zálohování, archivace, soubor, deduplikace, klient, server

Abstract

The purpose of this thesis is to design and create a backup and archiving data system for Company Vodovody a kanalizace Přerov, a.s. The work deals with development applications from the award itself, through its design to final implementation, deployment and testing operations. The system will consist of two main parts, namely the client and server. The client will allow backup files to a local store, compress, schedule and automatically run the backup process, synchronization, and connectivity to the central data server via LAN and restore data from backups that have been already created. Server will enable central management of backup storage, management of versions of stored data, data deduplication at the file level, the parallel connection of individual clients and their management, restoration of specific data to any location, and administrative routines for checking the database.

Keywords: backup, archive, file, deduplication, client, server

Seznam použitých zkratk a symbolů

AES	– Advanced Encryption Standard
FTP	– File Transport Protocol
ERD	– Entity Relationship Model
ORM	– Object-Relational Mapping
UDT	– User Defined Type
DFD	– Data Flow Diagram
MVC	– Model View Controller
GUI	– Graphical User Interface
UML	– Unified Model Language
NAS	– Network Attached Storage
DES	– Data Encryption Standard
SQL	– Structured Query Language
CVS	– Concurrent Version System
SVN	– Subversion
CRC	– Cyclic Redundancy Check
RAM	– Random Access Memory
XML	– Extensible Markup Language
WPF	– Windows Presentation Foundation
GFS	– Grandfather - Father - Son
GIS	– Geographic Information System
ODAC	– Oracle Data Access Components
NTFS	– New Technology File System
SŘBD	– Systém řízení báze dat

Obsah

1	Úvod	3
2	Požadavky zadavatele	4
2.1	Analýza současného stavu	4
2.2	Průzkum řešení na trhu	5
2.3	Volba výsledného řešení	10
3	Specifikace požadavků	11
3.1	Vstupy a výstupy systému	11
3.2	Definice funkčních specifikací	12
4	Analýza	14
4.1	Datová analýza	14
4.2	Funkční analýza	16
5	Návrh implementace	18
5.1	Specifikace implementačních nástrojů	18
5.2	Rozvržení modulů aplikace	19
5.3	Návrh systémových funkcí	22
5.4	Návrh vzhledu aplikace	28
6	Implementace	29
6.1	Implementace klienta	29
6.2	Implementace serveru	33
7	Závěr	34
7.1	Nasazení systému v praxi	34
7.2	Možná rozšíření systému	34
7.3	Závěrečné zhodnocení	34

Seznam obrázků

1	Norton Ghost Recovery Disc	5
2	Princip deduplikace Acronis	7
3	Use case diagram přístupu uživatelů	12
4	Use case diagram správy systému	13
5	ER diagram	15
6	DFD kontextové úrovně	16
7	DFD elementární úrovně	17
8	Package diagram rozvržení modulů aplikace	19
9	Návrh ORM	20
10	Architektura MVC	21
11	Sekvenční diagram zálohování na server	22
12	Schéma obnovy dat	24
13	Aktivitní diagram deduplikace při synchronizaci	26
14	Schéma zálohování GFS	27
15	Návrh GUI hlavního okna aplikace	28
16	Hlavní okno grafického uživatelského rozhraní	31
17	Okno nastavení aplikace	32
18	Informační okno aplikace	32

1 Úvod

Tato bakalářská práce se věnuje problematice zálohování a archivace dat v prostředí malých a středních firem. Jejím cílem je navrhnout a vytvořit systém manipulace s podnikovými daty, jejich správu a uchovávání, který kombinuje výhody centrálních datových skladů s benefity zálohování na lokální úložiště, a to s ohledem na použitelnost, jednoduchou správu, na nízké náklady pro pořízení, provoz a celkovou údržbu systému. Práce je rozdělena do jednotlivých částí, které odpovídají návrhu a vývoji celého řešení.

V druhé kapitole po tomto úvodu následuje kapitola Požadavky zadavatele, v níž je rozvedena analýza současného stavu zálohování a archivace ve společnosti zadavatele, seznámení s existujícími nástroji řešící danou problematiku a výběr konkrétního řešení. Nezanedbatelnou součástí průzkumu trhu je také finanční kalkulace, jež je jedním z hlavních aspektů při výběru výsledného řešení. Ve třetí kapitole jsou pak již specifikovány přesné požadavky na výsledný systém, jeho vstupy a výstupy a definice funkčních specifikací. Čtvrtá kapitola obsahuje strukturovanou analýzu systému, a to datovou a funkční. Nejobsáhlejší kapitolou je návrh implementace. Zde je specifikován výběr implementačních nástrojů, popsáno rozvržení jednotlivých modulů aplikace, jejich architektura, návrh systémových funkcí a vzhledu aplikace. Šestá kapitola, implementace, přibližuje detaily při psaní kódu serverové a klientské části systému a nastiňuje problémy, na než jsem při samotné implementaci narazil. Závěrečná sedmá kapitola pojednává o nasazení výsledného systému v praxi, možnostech jeho rozšíření a celkovém zhodnocení bakalářské práce.

Zadavatel, společnost Vodovody a kanalizace Přerov a.s. (Šířava 483/21, Přerov I. - město, 75002) je středně velká firma zabývající se výrobou pitné vody, zpracováním odpadních vod a údržbou kanalizací v Olomouckém kraji. Firma má několik poboček u technologických celků a dvě správní budovy v centru města Přerov, kde je soustředěna veškerá administrativní činnost a datové centrum. Požadavek zadavatele je právě řešení zálohování dat administrativní části v sídle společnosti.

2 Požadavky zadavatele

V první části této kapitoly jsem se zaměřil na analýzu současného stavu zálohování ve společnosti zadavatele, průzkum dostupných komerčních aplikací na trhu a na volbu výsledného řešení po domluvě se zadavatelem.

2.1 Analýza současného stavu

Pro důkladnou analýzu požadavků zadavatele a pro následný návrh jejich řešení je nutné analyzovat stávající stav zálohování a archivace dat ve společnosti. Topologii počítačové sítě zadavatele lze rozdělit na část datového centra a část osobních počítačů pracovníků administrativy.

2.1.1 Zálohování datového centra

Hardwarovou část datového centra společnosti tvoří BladeSystem od společnosti Hewlett Packard, který zahrnuje několik blade serverů a diskové pole. Důležitou vlastností tohoto systému je duplikace všech důležitých komponent nutných pro provoz zařízení a tím i zajištění jeho nepřetržitého běhu i v případě selhání některé z jeho komponent. Nad tímto hardwarovým zařízením je nainstalováno komplexní enterprise-level virtualizační prostředí VMware ESX, v němž se nachází 18 virtuálních serverů.

Zálohování všech virtuálních serverů zabezpečuje produkt společnosti Acronis, a to Backup and Recovery Advanced Server Virtual Edition. Celé řešení datového centra bylo dodavatelem koncipováno jako naprosto soběstačný a funkčně nezávislý celek, tudíž systém jeho zálohy je již hotov a není tak předmětem požadavků zadavatele.

2.1.2 Zálohování uživatelských stanic

Uživatelské stanice ve společnosti lze rozdělit do tří skupin. Tou největší je skupina osobních počítačů administrativních pracovníků, kterých je v současné době 48. Druhou skupinou jsou notebooky zaměstnanců společnosti, kteří velice často pracují v terénu. Trojici doplňuje malá skupinka speciálně určených pracovních stanic, jako jsou osobní počítače oddělení správy výpočetní techniky a workstation oddělení GIS.

Zálohování těchto stanic je řešeno kopírováním dat pomocí skriptů na jeden centrální počítač, z něhož jsou poté data nahrávána na páskovou jednotku, a to za použití schématu zálohování GFS (Grandfather-Father-Son). Aby byly zálohy uloženy na více fyzických míst současně a tím se předešlo možné ztrátě dat, přenáší skripty data ještě paralelně na několik úložišť NAS situovaných v obou budovách sídla společnosti.

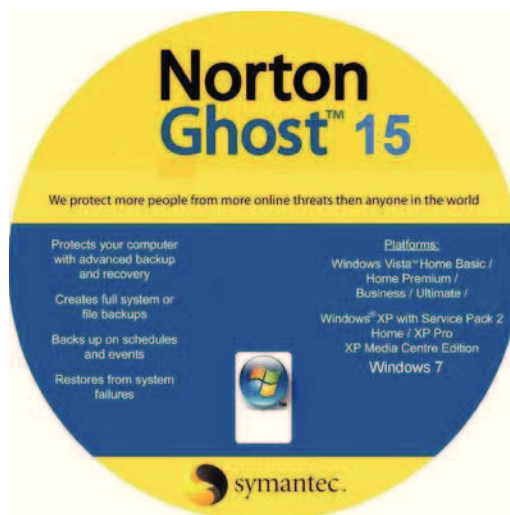
2.2 Průzkum řešení na trhu

V této kapitole jsou rozvedeny dnes nejznámější a nejpoužívanější komerční řešení na trhu. Průzkum zahrnuje jak aplikace sloužící k zálohování uživatelských dat na lokální úložiště pro nejjednodušší zálohu v prostředí domácností nebo těch nejmenších firem, tak robustní řešení centralizovaného zálohování pro střední a velké firmy.

2.2.1 Norton Ghost

Norton Ghost je jedním z nejznámějších produktů společnosti Symantec. Jedná se o profesionální program pro zálohu dat uložených na discích osobních počítačů. Umožňuje jak ruční, tak i automatické zálohování vybraných souborů a adresářů. Mnoho uživatelů jistě využije rychlé nastavení záloh, při kterém zvolí pouze data pro zálohování, cestu k adresáři pro ukládání a čas pro spuštění dané úlohy. V novějších verzích programu Norton Ghost je k dispozici i možnost využít jako datového úložiště FTP server nebo Network Attached Storage (NAS).

Pro předcházení následkům způsobených celkovou havárií operačního systému nabízí Norton Ghost možnost vytvoření bitové kopie disku, a to i bez nutnosti restartování počítače. V případě nutnosti obnovení celého disku z této kopie velmi dobře poslouží bootovací Symantec Recovery Disc, který lze vytvořit přímo z aplikace Ghost. Po startu operačního systému z tohoto disku lze připojit obnovovanou stanici k počítačové síti, najít úložiště obsahující zálohy a tyto poté obnovit.



Obrázek 1: Norton Ghost Recovery Disc

Díky funkci Recover my Files lze procházet všechny soubory a adresáře obsažené v libovolné záloze. Ty mohou tvořeny jak plnou zálohou, tak použitím inkrementální či

diferenciální zálohy. Další zajímavou funkcí je integrace kontroly zabezpečení Symantec ThreadCon, jež sleduje síťový provoz počítače a při detekci jakékoli hrozby napadení automaticky spustí zálohování.

Norton Ghost představuje ideální volbu pro zálohování v domácnostech nebo jako jednoduchý nástroj pro tvorbu bitových kopií systému v prostředí malých firem. Vzhledem k absenci funkce centralizovaného zálohování na server tento program navzdory svým kvalitám nevyhovuje požadavkům zadavatele a tudíž jej nelze použít jako možnou součást navrhovaného řešení.

Požadavky na systém:

- Windows XP SP2, Vista, 7
- 256MB RAM (512MB v případě použití recovery disku)
- 3GB místa na disku

Podporované souborové systémy pro zálohování:

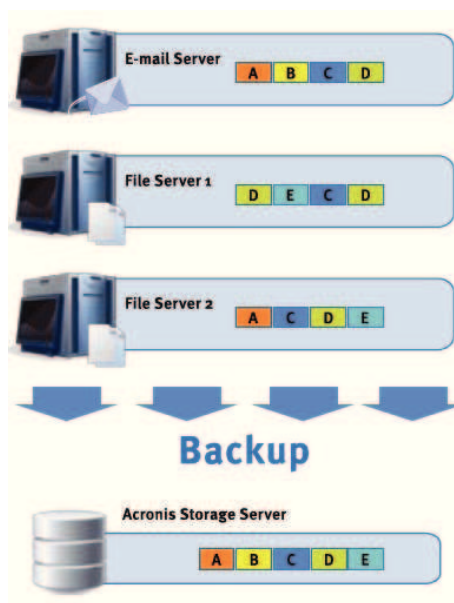
- Fat16, Fat32, NTFS
- Dynamické disky
- Linux Ext3

2.2.2 Acronis Backup and Recovery

Acronis Backup and Recovery Advanced Server je pokročilým systémem centralizovaného zálohování dat. Navazuje na úspěchy předchozí generace softwaru Acronis True Image a Acronis True Image Echo. Celý systém je velice robustní a vysoce sofistikovaný. Sám výrobce ve svých materiálech poukazuje na implementaci vlastních patentovaných technologií diskového pagingu. Cílovou skupinou zákazníků pro tento produkt jsou střední až velké firmy.

Podobně jako jeho předchůdce umožňuje zálohu a obnovu souborů, adresářů či celých disků včetně operačního systému běžících na platformě Windows i Linux, a to s důrazem na centralizaci těchto funkcí a dostupnost jejich konfigurace z jakéhokoli místa v síti. Produkt rovněž podporuje širokou škálu vizualizačních prostředí (VMware, Parallels, Microsoft Hyper-V), kdy umožňuje zálohu buď celého virtuálního stroje, nebo v něm obsaženého virtuálního disku.

Do již funkčního systému Advanced server je možno integrovat zásuvný modul Acronis Backup and Recovery Deduplication. Tento modul zajistí eliminaci ukládání stejných souborů z více připojených klientů, a tím ušetří nemalé náklady na pořízení a provoz datového úložiště. Kromě selektivní deduplikace ukládaných dat podporuje také deduplikaci na úrovni binárních bloků. Při kombinaci těchto dvou technologií spolu s kompresí výrobce proklamuje úsporu objemu dat až 90 procent. Výpočet operací deduplikace se provádí na klientovi, ovšem v případě omezení rychlosti jeho procesoru nabídne Acronis možnost provést tento výpočet až v cílovém Acronis Storage Node.



Obrázek 2: Princip deduplikace Acronis

Samotné propojení klientů se serverem splňuje velmi přísné požadavky na bezpečnost. Šifrované jsou jak všechna image záloh, tak i samotná síťová komunikace, a to 128-bitovým, 196-bitovým či 256-bitovým AES klíčem.

Acronis Backup and Recovery je velice zajímavou možností volby pro celkovou realizaci zálohování dat ve společnosti a jeho nasazení bude zváženo v analýze požadavků na výsledný systém.

Podporované operační systémy:

- Windows Server 2000, Windows 2003 a Windows 2008 (pro Advanced Server)
- Windows 2000 SP4, Windows XP SP2, Windows Vista a Windows 7 (pro klienty)

- Red Hat Enterprise 4 a 5, CentOS 4 a 5, Fedora 9 a 10, Ubuntu 8.10 a 9.04, Debian 4, SLES 10, OpenSUSE 11 a Asianux 3.0 (pro Advanced Server)

2.2.3 Symantec Backup Exec

Symantec Backup Exec je systém archivace dat pro středně velké firmy. Na rozdíl od konkurenčních řešení se orientuje kromě klasického zálohování souborů a složek na vyšší strukturovanost informací ve společnosti.

Technologie deduplikace dat je v produktu již plně obsažena a není nutno investovat do dalších rozšíření aplikace. Backup Exec dále umožňuje granulární obnovení databází Microsoft SQL, adresářů Active Directory či jednotlivé emaily ze systému Microsoft Exchange. Samozřejmostí u tohoto produktu je možnost obnovy celého operačního systému na jiný hardware či přímo jeho obnova do virtuálního prostředí Microsoft HyperV nebo VMware. Funkce Cold Backup umožňuje zálohu operačního systému bez nutnosti jeho startu.

V případě systému Symantec Backup Exec jde o přímou konkurenci řešení od společnosti Acronis. Na rozdíl od něj však nemá takovou tradici na trhu a přímé propojení na ostatní produkty Symantec (jako např. v případě spolupráce Acronis Backup and Recovery a Acronis True Image média pro obnovení). Nezanedbatelnou nevýhodou jsou také vyšší nároky na výkon hardware, které se v případě zakoupení antivirového modulu ještě zvyšují. Z těchto důvodů tento produkt nebude do celkové analýzy požadavků zahrnut.

Podporované operační systémy:

- Windows Server 2003, Windows Server 2008 R2 (pro server)
- Windows XP SP2, Windows Vista, Windows 7 (pro klienty)

Podporovaná virtuální prostředí:

- VMware vSphere 4.0
- Microsoft Hyper-V Server 2008 R2
- Citrix XenServer 5.x

2.2.4 Systémy verzování CVS a Apache Subversion

Verzování je způsob uchovávání historie veškerých provedených změn obecně u jakékoliv digitální informace. Nejčastěji se tento pojem používá u zdrojových kódů software, kdy se evidují změny provedené v jednotlivých verzích během stádia vývoje softwarového projektu. Verzovat lze všechny počítačové soubory, v případě potřeby se verzují i primární data. Z hlediska zálohy konkrétních uživatelských dat se zde přímo nabízí možnost návratu ke starší verzi souboru, než je poslední zálohovaná.

CVS (Concurrent Version System) je víceuživatelský systém typu klient - server používaný ke správě verzí dokumentů, nejčastěji pak celých projektů. Páteří celého systému je takzvaný repozitář (Repository). Ten obsahuje stromovou strukturu, v níž jsou uloženy jednotlivé projekty nebo skupiny těchto projektů. Repozitář také definuje přístup uživatelů k jednotlivým modulům v repozitáři uložených. Jeho obsah je pak uložen ve formě souborů na serveru.

Změny všech souborů repozitáře jsou sledovány a zaznamenávány na úrovni jednotlivých verzí. Díky stromové struktuře uchování dat je možné kdykoli a z kterékoliv verze vytvořit novou vývojovou větev a tu poté v případě potřeby opět sloučit s kořenovým projektem. Systém také umožňuje návrat k jakékoli předchozí verzi projektu či umožňuje publikaci různých vydání projektu.

Apache Subversion (dříve SVN) je v současné době jednou z náhrad za starší systém verzování CVS. Jeho nespornou výhodou je jeho volná dostupnost pro bezplatné komerční využití, kompletní dokumentace a otevřené zdrojové kódy, lze jej provozovat na platformách Windows i Linux.

Subversion je vhodné použít všude tam, kde je kladen vysoký důraz na evidenci a správu změn dokumentů, které jsou často upravovány ať již jedním či více uživateli. Výstižným příkladem je aplikace takového systému v softwarové firmě, kde vývojové týmy spolupracují na různých projektech. I když lze v podstatě verzovat jakýkoli soubor, tak se SVN používá především v prostředích s vysokým podílem souborů s textovým nebo vnitřně strukturovaným obsahem, jako jsou textové dokumenty, dokumenty elektronické pošty, softwarové projekty atp. Naopak méně vhodné je jejich nasazení v prostředích s vysokým počtem souborů v binárním tvaru, které nejsou téměř vůbec modifikovány a tudíž zde ztrácí verzování svůj smysl.

2.3 Volba výsledného řešení

Z analýzy současného stavu zálohování ve společnosti jasně vyplývá, že stávající řešení je vzhledem k relativně vysokému počtu uživatelů a tím i k strmě vzrůstajícím požadavkům na režii celého zálohování naprosto nevyhovující. Případná obnova dat ze záloh je zdlouhavá a vysoce časově náročná, přehled o stavu zálohování jednotlivých stanic je nevalný.

Jako řešení problému se nabízela možnost zakoupení rozšíření pro produkt Acronis Backup and Recovery. Vzhledem k vysoké finanční náročnosti při použití tohoto produktu dle finanční kalkulace, jeho nadbytečné robustnosti pro zálohování dat běžných uživatelů a složitému uživatelskému rozhraní bylo od nákupu tohoto řešení upuštěno.

Cenová kalkulace pro zálohování cca 50 osobních počítačů a 10 notebooků by zahrnovala:

- Acronis Backup and Recovery Advanced Server - 25 172 Kč.
- Acronis Backup and Recovery Universal Restore - 15 680 Kč.
- 60 x Acronis Backup and Recovery Advanced Workstation + Deduplication for Advanced Workstation - 60x 2 716 Kč.
- Celkem tedy 203 812 Kč bez hardware (ceny k 4.1.2011, zdroj AutoCont.cz).

Dalším řešením byla možnost aplikace systému správy verzí. Po konzultaci s uživateli se ukázalo, že většinu dat uživatelů tvoří soubory, u nichž se provádí změny v minimální míře. Převažují mezi nimi fotografie technických celků ve správě společnosti a dokumenty ve formátu pdf vytvořené z vnějšku firmy zadavatele (nejčastěji jde o mapové podklady či směrnice různá pro dané oddělení). Vnitropodnikovou poštu, stejně jako systém obědnávek administrační části společnosti zajišťují systémy třetích stran, tudíž jejich zálohování není nutno řešit.

S přihlédnutím k těmto faktům a po poradě se zadavatelem byla zvolena možnost vyvinout vlastní systém zálohování, který je předmětem této bakalářské práce.

3 Specifikace požadavků

V této kapitole bude podrobně rozvedana specifikace požadavků na výsledné řešení, popis jeho vstupů a výstupů, funkcionality a modely funkční specifikace pomocí případů užití.

3.1 Vstupy a výstupy systému

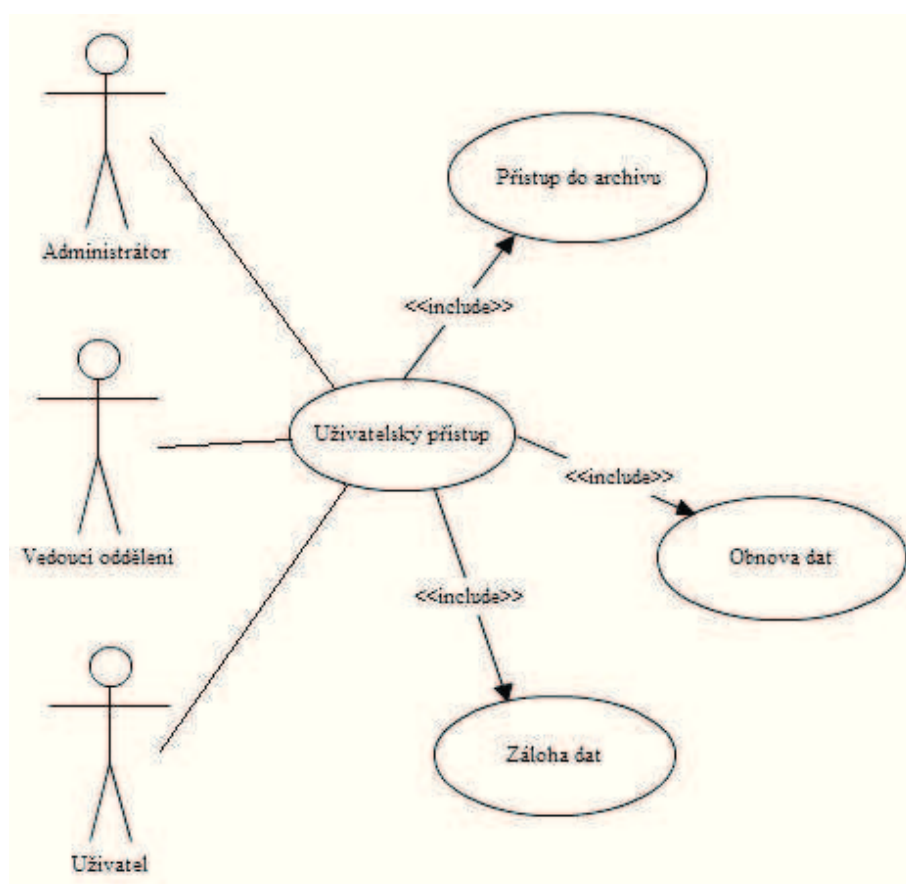
Předmětem požadavků zadavatele je systém pro centralizované zálohování uživatelských dat jednotlivých pracovních stanic. Aplikace by měla splňovat tyto požadavky:

- Klientská aplikace bude umožňovat zálohování uživatelských dat na jeden centrální server.
- Celý proces bude automatizován, kdy bude možnost nastavit dobu spuštění zálohovací procedury pro každého uživatele.
- Data jednotlivých záloh bude možné procházet jako adresářovou strukturu, bude možné obnovit jak kompletní zálohovanou sestavu, tak jakýkoli uzel adresářové struktury (soubor nebo celou složku).
- Aplikace bude umožňovat vytvoření kompletní zálohy na lokální úložiště s využitím komprese pro zachování systému zálohování na více míst současně.
- Každý uživatel bude mít možnost odeslat důležitá data určená k archivaci (výroční zprávy, uzávěrky atd.) do datového archivu, zobrazovat obsah tohoto archivu a stahovat z něj informace dle přidělených uživatelských práv.
- Prostředí aplikace musí být intuitivní, přehledné a snadno pochopitelné pro běžné uživatele.
- Systém zálohování musí být včlenitelný do stávající topologie sítě společnosti a měl by pokud možno co nejvíce využít stávajícího hardwaru z důvodu úspory finančních prostředků.

3.2 Definice funkčních specifikací

Nedílnou součástí specifikace požadavků je UML diagram případů užití Use Case Diagram. Jeho účelem je definice chování systému (případů užití) a popis jeho vnějšího okolí (aktérů).

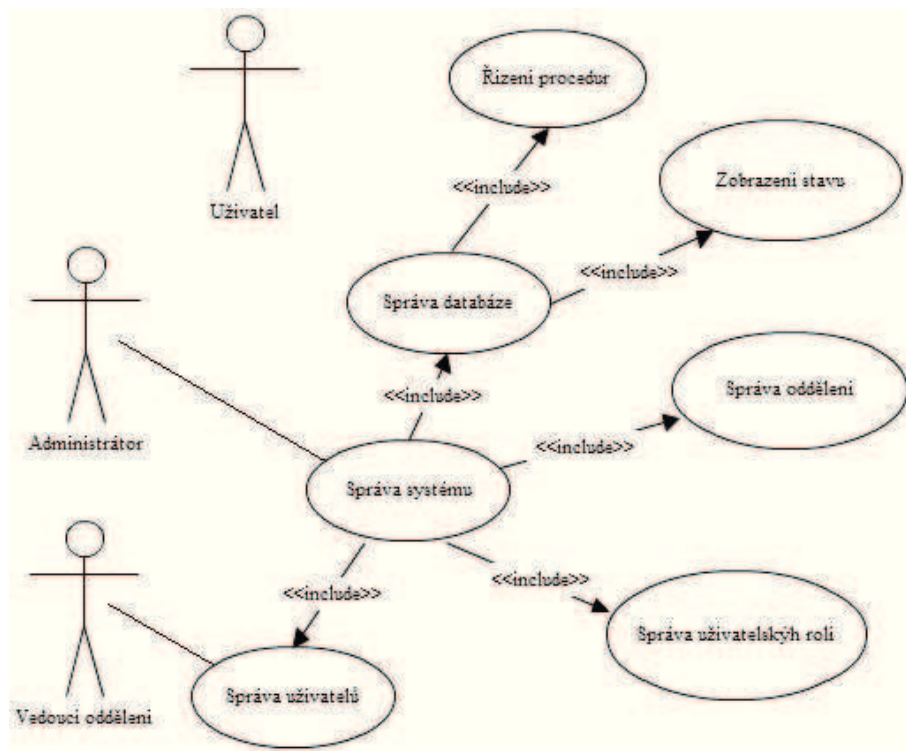
Diagram přístupu uživatelů specifikuje možnosti využití hlavních funkcí systému jednotlivých uživatelů. Ty musí být k dispozici jak administrátorovi, tak vedoucím oddělení a běžným uživatelům. V případě nedostupnosti centrálního serveru se klientský modul automaticky spustí v offline režimu, přičemž musí stále nabízet možnost zálohování a obnovení na lokální úložiště.



Obrázek 3: Use case diagram přístupu uživatelů

Diagram Správa systému poskytuje přehled v přístupu jednotlivých aktérů k nastavení a řízení systému. Práva pro správu systému bude mít pouze osoba přihlášená jako administrátor. Jemu je umožněno spouštět optimalizační procedury databáze, zobrazit její aktuální stav a modifikovat uživatelské účty, uživatelské role a oddělení. Pokud bude

uživatel přihlášen jako vedoucí oddělení, bude mu umožněna správa uživatelských účtů. Běžnému uživateli nebude povolena žádná možnost správy systému.



Obrázek 4: Use case diagram správy systému

4 Analýza

Strukturovaná analýza je jednou z možností, jak lze složitý návrh systému zpřehlednit. Při analýze systému je zapotřebí vytvořit několik modelů, jež reprezentují obecnou podobu reálného systému bez přihlédnutí k pozdější implementaci. Tyto modely zahrnují charakteristiku všech vlastností jednotlivých komponent systému a popis událostí a interakcí mezi nimi. Modely analýzy se z pohledu systému dělí na dvě základní skupiny, a to na datovou a funkční analýzu.

4.1 Datová analýza

Datová analýza popisuje databázovou část vyvíjeného softwarového systému. Eviduje objekty a jejich atributy, určuje funkční závislosti a definuje datové struktury. Součástí datové analýzy je lineární zápis typů entit, popis atributů, datový slovník a ER diagram.

4.1.1 Lineární zápis typů entit

Lineární zápis typů entit specifikuje entitní typy a vztahy mezi nimi. Tyto vztahy jsou realizovány pomocí klíčových atributů, jež tvoří primární a cizí klíče jednotlivých entit. V následujícím zápisu je primární klíč zvýrazněn tučně, cizí klíč zapsán kurzívou.

Oddeleni(**id_oddeleni**, *nazev_oddeleni*)

Uzivatel'ska_Role(**id_role**, *nazev_role*, *menit_oddeleni*, *menit_uzivatel*, *menit_role*, *videt_data_skupina*, *videt_data_pc*)

Uzivatel(**id_uzivatel**, *jmeno*, *prijmeni*, *username*, *id_oddeleni*, *id_role*, *cas_synchro*, *diskova_kvota*)

Hesla(**id_hesla**, *id_uzivatel*, *heslo*)

Zalohovane_Sestavy(**id_sestavy**, *id_uzivatel*, *datum*, *sestava*)

Binarni_Data(**id_data**, *crc*, *velikost*, *počet_odkazu*, *datum_nahrani*, *jmeno*, *typ*, *ulozena_data*)

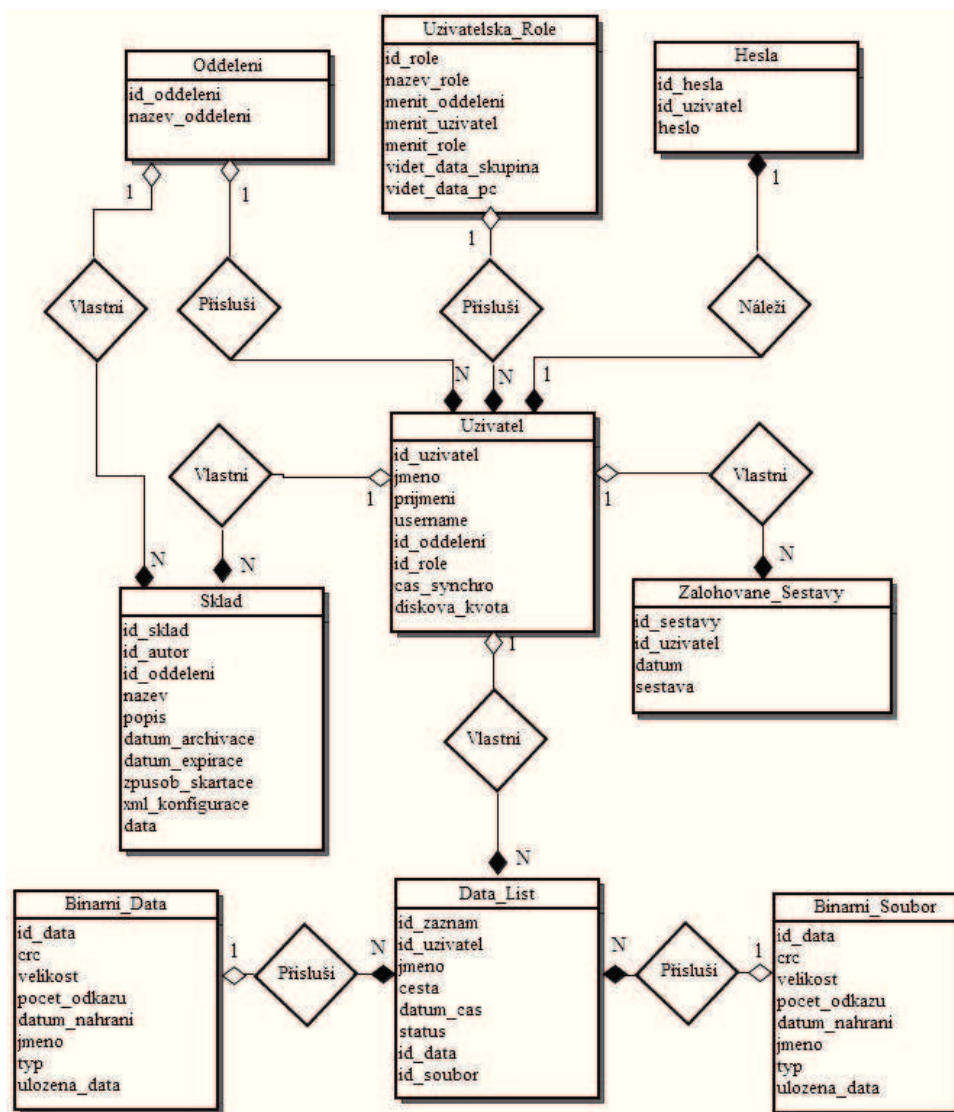
Binarni_Soubor(**id_data**, *crc*, *velikost*, *počet_odkazu*, *datum_nahrani*, *jmeno*, *typ*, *ulozena_data*)

Data_List(**id_zaznam**, *id_uzivatel*, *jmeno*, *cesta*, *datum_cas*, *status*, *id_data*, *id_soubor*)

Sklad(**id_sklad**, *id_autor*, *id_oddeleni*, *nazev*, *popis*, *datum_archivace*, *datum_expirace*, *způsob_skartace*, *xml_konfigurace*, *data*)

4.1.2 ER Diagram

ER Diagram se používá pro konceptuální znázornění entit, jejich typů a vazeb. Entitami se rozumí objekty reálného světa či pojmenování akce a jsou označovány podstatnými jmény.



Obrázek 5: ER diagram

Vztah (Relationship) je označení, jakým způsobem jsou dvě nebo více entit vztaženy mezi sebou. Vztahy jsou označovány slovesy. Povinnost členství ve vztahu je označována symbolem na konci čar entity spojujících. Prázdný symbol značí nepovinnost ve vztahu, plný naopak povinnost ve vztahu. Počet k sobě náležících entit označujeme jako vazby. Jejich výčet obsahuje následující tabulka:

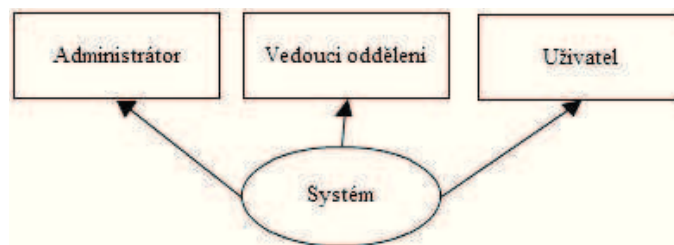
- 1:1 - Jedné entitě náleží právě jedna entita druhého typu
- 1:N - Jedné entitě náleží jedna nebo více entit druhého typu, ne však naopak
- M:N - Jedné entitě náleží jedna nebo více entit druhého typu a naopak

4.2 Funkční analýza

Funkční analýza je popisem funkcí systému a jejich vzájemných závislostí přičemž vychází z požadovaných vstupů, výstupů a funkcí zadaných ve specifikaci požadavků systému. Výsledkem této analýzy je funkční model, který obsahuje znázornění datových toků konkrétních funkcí a logiku jejich algoritmizace.

4.2.1 Diagram datových toků

Diagram datových toků vytváří názorný pohled na systém a jeho funkce. Má hierarchickou strukturu, přičemž lze celý diagram rozložit na menší části a ty zvlášť rozkreslit, kdy se postupuje směrem shora dolů. Rozlišujeme úrovně vrchní (kontextovou), střední a koncovou (elementární). Nejmenší systémy lze reprezentovat jedním DFD.



Obrázek 6: DFD kontextové úrovně

Základními prvky diagramu datových toků jsou:

- funkce - označeny elipsou
- paměť - označeny dvěma vodorovnými čarami
- aktéři - označeni obdélníkem
- datové toky - označeny šipkou

5 Návrh implementace

Účelem návrhu implementace je doplnění analýzy projektu o specifikaci typů implementačních nástrojů, systémových funkcí a organizačních operací systému. Jeho součástmi jsou:

- Specifikace implementačních nástrojů
- Diagram rozvržení modulů aplikace
- Návrh systémových funkcí
- Návrh vzhledu aplikace

5.1 Specifikace implementačních nástrojů

Klientská část systému se skládá z vlastního jádra aplikace a z grafického uživatelského rozhraní. Pro samotnou implementaci jsem zvolil platformu Microsoft .NET verze 4.0, programovací jazyk C# a vývojové prostředí Microsoft Visual Studio 2010.

Microsoft .NET Framework je nejrozšířenější platforma pro osobní počítače s operačním systémem Microsoft Windows. Podporuje celou řadu programovacích jazyků (například C#, J#, C++), přičemž výsledná aplikace není přeložena přímo do zdrojového kódu, nýbrž do mezijazyka Common Intermediate Language. Tato vlastnost činí aplikaci snáze přenositelnou mezi různými operačními systémy nebo dokonce i platformami, stačí když hostitelský operační systém podporuje .NET Framework dané verze.

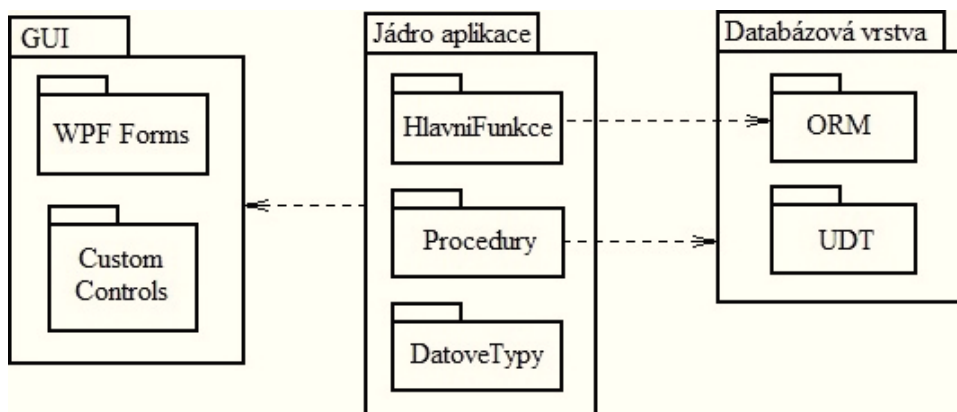
Microsoft Visual Studio 2010 je robustní vývojové prostředí použitelné pro vývoj jak desktopových, tak i mobilních a webových aplikací na platformách Microsoft Windows, Windows Mobile, Windows CE, .NET a Silverlight. Mezi jeho hlavní výhody patří pracovaný systém automatického doplňování kódu IntelliSense, refraktorování, vizuální designer a široká škála nástrojů, například pro práci s databázemi či práci ve vývojovém týmu.

Serverová část systému obsahuje databázový stroj, který umožňuje ukládání dat, jejich evidenci a definuje procedury a funkce pro manipulaci s nimi.

Pro vlastní databázi jsem vybral Oracle Database v11g. Jedná se o systém řízení báze dat (SŘBD), který podporuje relační dotazovací jazyk SQL, objektové databáze a hierarchický model dat. Jako vývojové prostředí pro realizaci jsem zvolil Oracle SQL Developer a Microsoft Visual Studio s doplňkem Oracle Data Access Components.

5.2 Rozvržení modulů aplikace

Rozvržení jednotlivých modulů aplikace je znázorněno diagramem seskupení (Package Diagram). Ten ukazuje skupiny, ve kterých jsou organizovány jednotlivé příbuzné elementy systému, a závislosti mezi nimi. Používá se jako metoda pro uspořádání rozsáhlejších systémů. Mechanismus seskupování lze aplikovat na libovolné elementy a je možné vytvářet jejich víceúrovňové hierarchické struktury.



Obrázek 8: Package diagram rozvržení modulů aplikace

5.2.1 Jádru aplikace

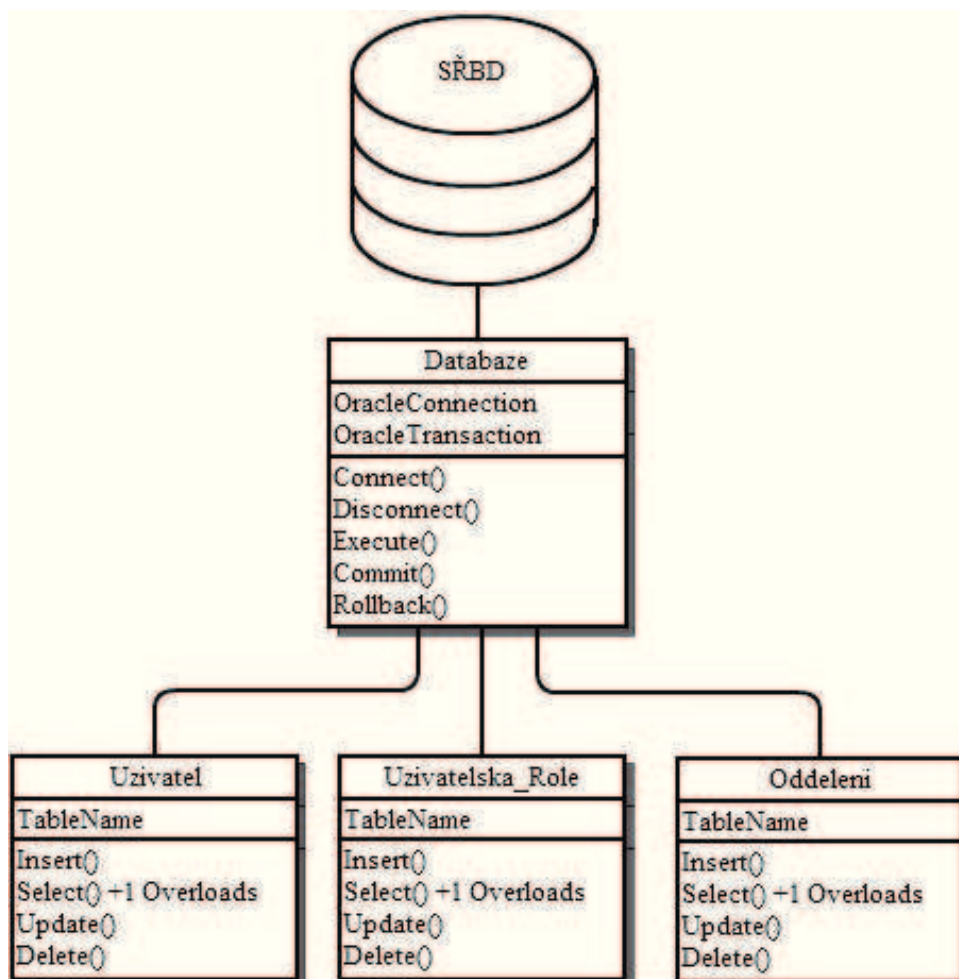
Jádru aplikace v sobě zapouzdřuje veškerou funkčnost klientské části systému. Obsahuje hlavní procedury pro práci se soubory, kryptografií, přenos dat přes počítačovou síť, moduly pro automatické spouštění procesu zálohování a notifikace o jeho průběhu pomocí emailu.

5.2.2 Databázová vrstva

Databázová vrstva v sobě zapouzdřuje oběktově relační model přístupu k databázi. ORM je technika, která zajišťuje konverzi dat mezi objektovým a relačním datovým modelem. Díky tomu je vývojář při práci s daty, které jsou v aplikaci reprezentovány pomocí objektů, do jisté míry zbaven nutnosti pracovat s SQL dotazy relační databáze. ORM zejména usnadňuje provádění základních operací výběru, vkládání, editace, mazání dat a zajišťuje řízení transakcí při přístupu do databáze. Velikou výhodou ORM je také konverze datových typů mezi databází a programovacím jazykem.

V návrhu aplikace jsou však metody, u kterých je požadováno, aby se připojovaly k procedurám databáze napřímo bez použití jakékoli mezivrstvy. Jedná se hlavně o metody pro upload souborů, kdy je jejich management transakcí řízen samotným SRBD. Pro zjednodušení komunikace s databází jsem při návrhu využil uživatelsky definovaných

typů, díky nimž lze procedurám SŘBD předávat jako parametr objekty či tabulky těchto objektů.



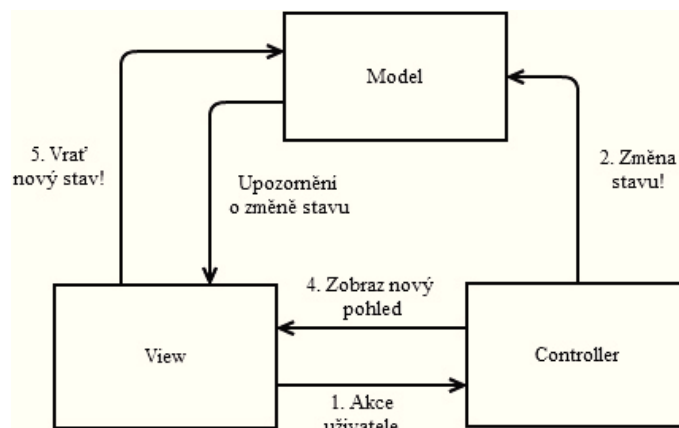
Obrázek 9: Návrh ORM

5.2.3 Grafické uživatelské rozhraní

Grafické uživatelské rozhraní slouží jako komunikační rozhraní mezi uživatelem a aplikací. Pro propojení aplikace s GUI se používá návrhový vzor Model-View-Controller (zkráceně MVC). Jak již název vypovídá, architektura MVC se skládá ze tří částí:

- Model se skládá množiny tříd a funkcí které provádějí manipulaci s daty.
- View představuje uživatelské rozhraní aplikace. Mohou to být prvky GUI, například GridView aj.

- Controller naslouchá událostem přicházejícím od uživatele (prostřednictvím View) a rozhoduje, jaké metody Modelu se budou volat jako odpověď na akci uživatele.



Obrázek 10: Architektura MVC

Samotná činnost aplikace vypadá následovně:

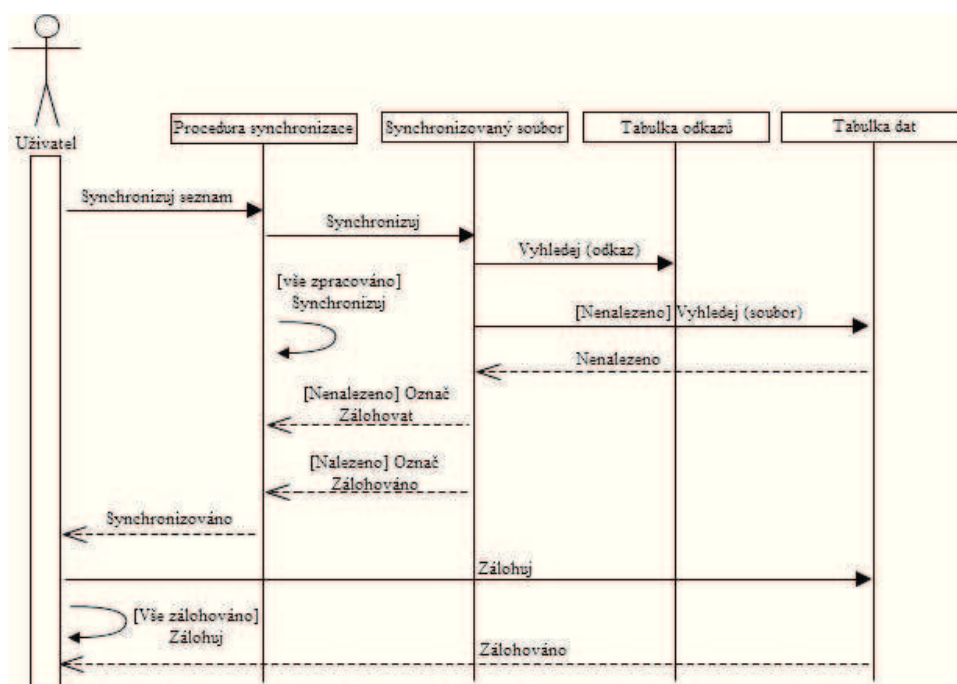
1. Uživatel provede akci (stiskne tlačítko, označí položku formuláře apod.)
2. View předá informace Controlleru, který informuje Model o změně
3. Model provede aktualizaci vnitřního stavu dat
4. Controller dá pokyn View pro zobrazení nového pohledu
5. View zobrazí aktuální data z Modelu
6. Je očekávána další akce uživatele a celý cyklus se opakuje

5.3 Návrh systémových funkcí

V této kapitole je podrobně vysvětlen návrh všech klíčových funkcí systému tak, jak budou později implementovány.

5.3.1 Zálohování dat na server

Proces zálohování dat na server začíná výběrem souborů a složek provedeným klientem. Tento výběr může být také uložen v konfiguračním XML souboru a při zahájení procesu zálohování pouze načten. Jakmile je sestava pro zálohování načtena, přechází se ke kroku zjištění informací o aktuálně přihlášeném uživateli. V tomto kroku je velice důležitý jednoznačný identifikátor daného uživatele (ID), podle nějž bude server přistupovat do databáze. Po tomto kroku je spuštěna procedura zpracování zálohované sestavy, v jejímž průběhu jsou zjištěny klíčové informace o každém zálohovaném souboru, jako je jeho jméno, úplná cesta, velikost a kontrolní součet CRC. Tyto údaje jsou zapsány do synchronizační sestavy, která je v dalším kroku zaslána na server ke zpracování. Server přijme celý seznam dat, porovná jej s databází, označí nezálohované soubory a odešle jej klientovi zpět. Po obdržení seznamu klientem tento začne uploadovat nezálohovaná data na server. Až jsou všechna data úspěšně přenesena, klient vytvoří XML soubor reprezentující adresářovou strukturu zálohovaných souborů s údaji pro jejich obnovu. Tento soubor slouží jako bod obnovení a je uložen na centrální server. Po tomto kroku je zálohování na server ukončeno.



Obrázek 11: Sekvenční diagram zálohování na server

5.3.2 Zálohování dat na lokální úložiště

Podobně jako při zálohování dat na server, i při zálohování na lokální úložiště začíná celý proces výběrem souborů a zpracováním informací o nich. Po tomto kroku je zkontrolována dostupnost nastavené cesty k lokálnímu úložišti a ověřena jeho kapacita. Pokud je úložiště připraveno, započne komprese dat a jejich přenos. Po úspěšném dokončení přenosu je vygenerován bod obnovení s informacemi o obsahu záložního archivu a uložen jako XML soubor v cílovém lokálním úložišti.

5.3.3 Archivace dat

Jednou z funkcí aplikace je také možnost uživatelů odeslat vybranou množinu souborů do archivní části systému. Tato část je specifická tím, že data z ní nelze vymazat jiným způsobem, než procedurou odstraňující data po uplynutí doby jejich expirace nebo zásahem administrátora. Poté, co uživatel vybere soubory k archivaci, je vyzván k vyplnění názvu archivovaného balíčku, jeho popisu, datu expirace a způsobu jejich skartace. Poté je takto podepsaný balíček zkomprimován a odeslán na server. Jeho obsah je uložen jako XML soubor umožňující zobrazit jeho obsah jako adresářovou strukturu.

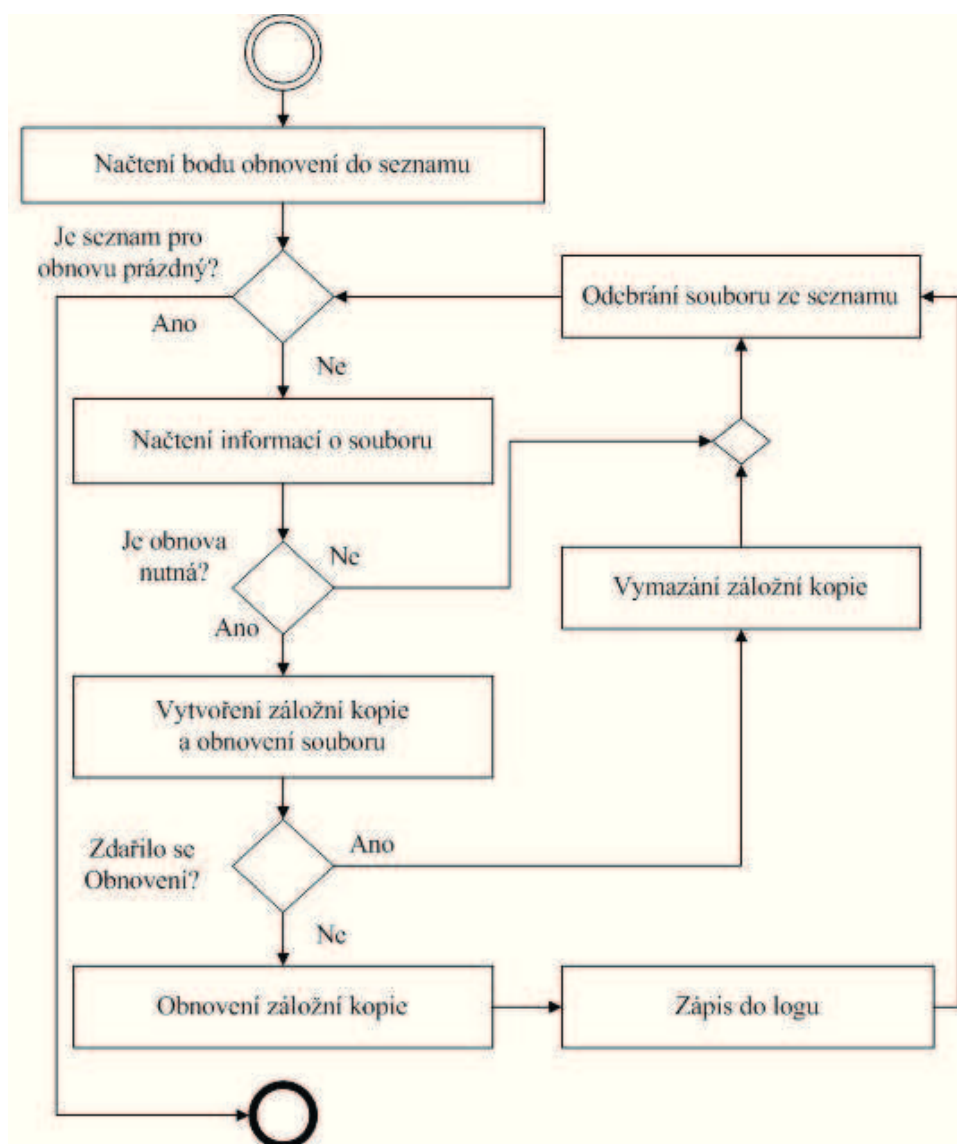
5.3.4 Obnova dat ze zálohy

Proces obnovy dat ze zálohy je velice podobný jak v případě obnovy ze serveru, tak v případě obnovy z lokálního úložiště. Po načtení bodu obnovení se vytvoří synchronizační tabulka podobná té, jež se tvoří při procesu zálohování. V tomto případě je však celý proces reverzní. Pro každý záznam tabulky, jež reprezentuje jeden soubor, se zjistí, jestli již není v cíli přítomen. Pokud ano, porovná se, zdali je soubor stejný či ne. Pokud ano, soubor se neobnovuje, což zvýší výkon celé procedury obnovení. Pokud není stejný, je vytvořena jeho záložní kopie a započata obnova původního obsahu. Po úspěšném obnovení je záložní kopie vymazána a pokračuje se dalším souborem ze seznamu.

5.3.5 Princip deduplikace dat při synchronizaci

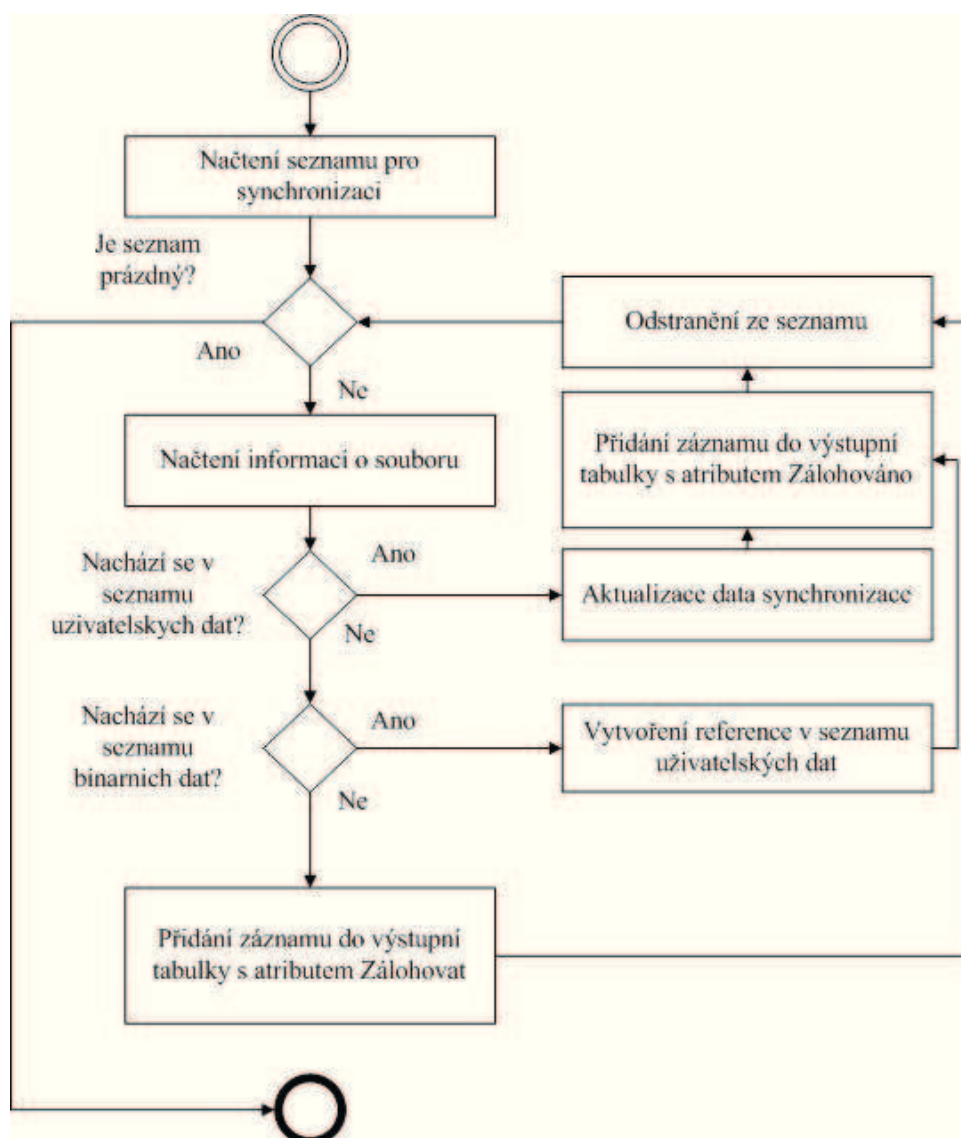
Deduplikace Deduplikace je technologie odstranění duplicitních dat z jedné zálohy nebo i z více záloh dohromady. Lze ji uplatnit i na rozdílové a inkrementální zálohy. Rozlišujeme aplikaci deduplikace na úrovni souborové a úrovni binárních dat (blokové).

Při deduplikaci na úrovni souborů je uložen originální soubor, a při duplicitách se ukládají pouze odkazy na něj. Při větším počtu klientů tak lze ušetřit velké množství kapacity, a v neposlední řadě i náklady na pořízení a provoz takového systému. Při zálohách souborů s velikostí v řádech GB se používá deduplikace na úrovni dat. Tato metoda ukládá pouze ty bloky dat, které jsou rozdílné od originálu. Při následné obnově je potom nutné data zpětně rekonstruovat. Tato metoda se nejčastěji využívá pro zálohování mohutných virtuálních prostředí, např. VMware ESX. Rozlišujeme také deduplikaci na zdroji (klientovi), a cíli (serveru). Výhodou deduplikace na zdroji je, že systém detekuje redundantní data již před přenesením na úložiště, čímž minimalizuje objem dat přenášených přes síť.



Obrázek 12: Schéma obnovy dat

Návrh deduplikace Při spuštění procesu zálohování je z vybraných souborů a adresářů vytvořen synchronizační seznam obsahující klíčové atributy jednotlivých souborů, jako je jméno, úplná cesta, velikost a hash. Tato sestava je poté odeslána na server. Při zpracovávání synchronizačního seznamu serverem je každý záznam porovnán s databází uložených odkazů na binární soubory aktuálního uživatele. Pokud je ukládaný soubor stejný a je na něj vytvořena reference, není potřeba tento soubor dále zálohovat. Aktuálně zpracováváný záznam synchronizační tabulky je tudíž označen jako zálohovaný. Pokud ukládaný soubor nemá referenci uživatele, avšak jeho binární obsah již uložený v databázi je, je vytvořena reference a soubor je tak automaticky zálohován. V opačném případě je nutné soubor zálohovat, a tudíž je jeho záznam označen jako nezálohovaný. Poté je celý seznam odeslán zpět klientské stanici, která začne uploadovat pouze ty soubory, jež v databázi nejsou obsaženy.

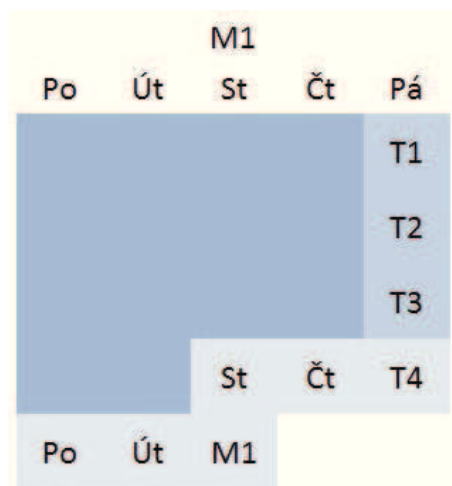


Obrázek 13: Aktivitní diagram deduplikace při synchronizaci

5.3.6 Volba zálohovacího schématu

Z principu kumulace nových záloh v systému vyplývá, že nelze uchovávat libovolné množství různých verzí těchto záloh. Proto je nutno přesně zpracovat metodiku, jak bude celý proces tvorby záloh probíhat. Ve firemním prostředí dochází k problému zpracování velkých datových objemů, což může vést k přetěžování síťové infrastruktury a datových úložišť. Z těchto důvodů se vytváří pravidelný časový plán, který pomůže důsledky těchto problémů výrazně omezit a umožní dostatečný přehled a kontrolu nad průběhem celého procesu zálohování. Tato metodika je také někdy nazývána jako plán rotace záloh, a vychází z principů rotace datových pásek v minulosti.

Nejběžněji používané schéma zálohování, které jsem také vybral pro účely tohoto systému, je metoda Grandfather-Father-Son. Tohle schéma využívá tři typy media setů. Čtyři media sety označené jako Son pro každodenní zálohu v pracovním týdnu (pondělí až čtvrtek) s dobou expirace jeden týden, pět media setů Father pro týdenní zálohy s dobou expirace jeden měsíc a tři media sety Grandfather pro měsíční zálohu s expirací tři měsíců.



Obrázek 14: Schéma zálohování GFS

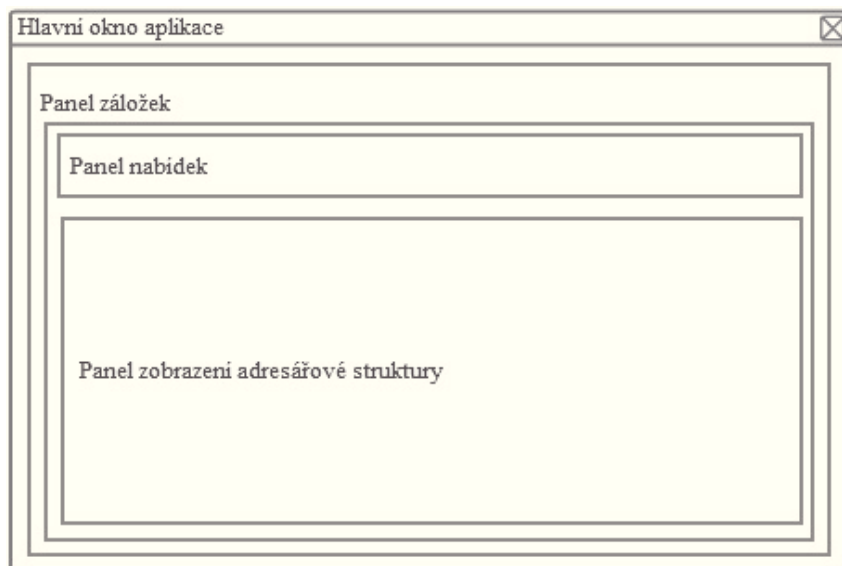
Principem je rotace těchto media setů, kdy každý týden jsou denní zálohy uloženy na media sety Son, v pátek každého týdne se provede záloha na media set Father a každý první týden v měsíci se provede záloha na media sety Grandfather. Tohle schéma zabezpečuje možnost návratu k verzím o den, týden a měsíc zpět. Po vypršení doby expirace každého media setu je tento znovu přepsán v dalším rotačním cyklu záloh.

5.4 Návrh vzhledu aplikace

Vzhled výsledné podoby aplikace je součástí specifikace požadavků. Vhodnost jeho návrhu je základním kritériem hodnocení kvality aplikace od běžných uživatelů. Při práci s GUI je důležité, aby si uživatel vytvořil správný mentální model aplikace, a to co nejrychleji. Toho lze dosáhnout:

- jednotným stylem grafického zpracování
- utříděným a dobře prezentovaným obsahem
- intuitivitou ovládání
- vhodnou volbou barev

Po konzultaci se zadavatelem jsem navrhl čistý minimalistický vzhled aplikace čerpající ze vzhledu dialogů, které jsou dispozici při použití technologie Windows Presentation Foundation. Hlavní okno, po vzoru moderních aplikací, postrádá hlavní menu a stavový řádek, veškeré volby zprostředkovává panel záložek, jež je hlavním stavebním prvkem celého okna. Jednotlivé záložky poté obsahují panel tlačítek pro výběr jednotlivých funkcí a panel zobrazení adresářové struktury lokálních disků či záloh. Je tak zaručen jednotný vzhled a rychlá orientace v aplikaci. Pozadí jsem zvolil lehce modré přecházející do bílé barvy z palety tzv. bezpečných barev web-safe colors. Dialogy upozornění a chyby jsou pro lepší rozlišení s červeným pozadím.



Obrázek 15: Návrh GUI hlavního okna aplikace

6 Implementace

Implementace je dalším krokem životního cyklu vyvíjené aplikace. Jejím cílem je realizovat navrženou architekturu pomocí programového kódu a vytvořit tak ucelený funkční systém. Implementace klade důraz na fyzickou organizaci komponent dle implementačního prostředí a zvolených programovacích jazyků.

6.1 Implementace klienta

V této části kapitoly jsem uvedl příklady řešených problémů, se kterými jsem se setkal při implementaci, použitých technologií při řešení těchto problémů a ukázkou implementace grafického uživatelského rozhraní.

6.1.1 Hashovací funkce

Prvním problémem při implementaci aplikace pracující se soubory je nutnost vyřešit jednoznačnou identifikaci každého konkrétního souboru. Pokud s nimi pracujeme jako s binárními daty, je na první pohled zřejmé, že pouhá identifikace podle cesty, názvu a ostatních atributů postačovat nebude. Proto je nutné použít hashování funkce.

Hashovací funkce je matematická procedura, která vytvoří z dat na jejím vstupu určitou krátkou posloupnost čísel, která je jejich stopou, otiskem (anglicky Fingerprint). Hlavní vlastností této funkce je fakt, že výstupní posloupnost má konstantní délku nezávisle na délce vstupu, přičemž i malá změna na vstupu způsobí velkou změnu na výstupu. Do jisté míry lze říci, že výstup hashovací funkce je pro jakákoli vstupní data unikátní.

Jelikož má hashování funkce pevnou délku otisku, existuje zde jistá teoretická pravděpodobnost, že pro dva různé soubory bude výsledný hash kód stejný. Z tohoto důvodu je jednoznačná identifikace souboru v systému zajištěna kombinací hashovacího algoritmu MD5 a velikostí daného souboru v bytech.

6.1.2 Kryptografie TripleDES

Základním požadavkem na bezpečnost citlivých dat je nutnost implementace kryptografie, a to ať již pro uchovávání hesel aplikací, tak pro šifrování jednotlivých bloků přenášených dat. TripleDES je bloková šifra založená na symetrické šifře DES (Data Encryption Standard). V současné době se používá například pro šifrování SSL spojení. Principem algoritmu je trojnásobná aplikace původního algoritmu DES za sebou. Výsledný klíč má takto délku 168 bitů.

```
public byte[] zasifruj (byte[] vstup, string klic, string heslo)
{
    byte[] output = new byte[0];
    TripleDES td = TripleDES.Create();
    td.Key = Encoding.ASCII.GetBytes(klic);
    td.IV = Encoding.ASCII.GetBytes(heslo);
    td.Mode = CipherMode.CBC;
    td.Padding = PaddingMode.Zeros;
    ICryptoTransform ict = td.CreateEncryptor();
    output = ict.TransformFinalBlock(vstup, 0, vstup.Length)
}

public byte[] desifruj (byte[] data, string klic, string heslo)
{
    byte[] output = new byte[0];
    TripleDES td = TripleDES.Create();
    td.Key = Encoding.ASCII.GetBytes(klic);
    td.IV = Encoding.ASCII.GetBytes(heslo);
    td.Mode = CipherMode.CBC;
    td.Padding = PaddingMode.Zeros;
    ICryptoTransform ict = td.CreateDecryptor();
    output = ict.TransformFinalBlock(data, 0, data.Length);
    return output;
}
```

Výpis 1: Ukázka kryptografické funkce v jazyce C#

6.1.3 Notifikace emailem

Při implementaci klientské části aplikace byl zahrnut předpoklad, že po většinu času bude program samotný běžet skrytý v oznamovací oblasti GUI operačního systému a veškerá činnost zálohování bude automatizována. To ovšem znamená, že uživatel ztrácí přehled o tom, zdali zálohování proběhlo úspěšně. Pokud se tak nestalo, je nutné uvědomit administrátora systému, který sjedná nápravu daného problému.

Tento nedostatek je vyřešen automatickým zasíláním emailu na pevně danou emailovou adresu. Díky tomu je správce systému vždy informován o aktuálním stavu záloh v celé společnosti. Ukázka implementace včetně ošetření výjimky v případě nedostupnosti připojení k poštovnímu serveru je patrná na následujícím kódu:

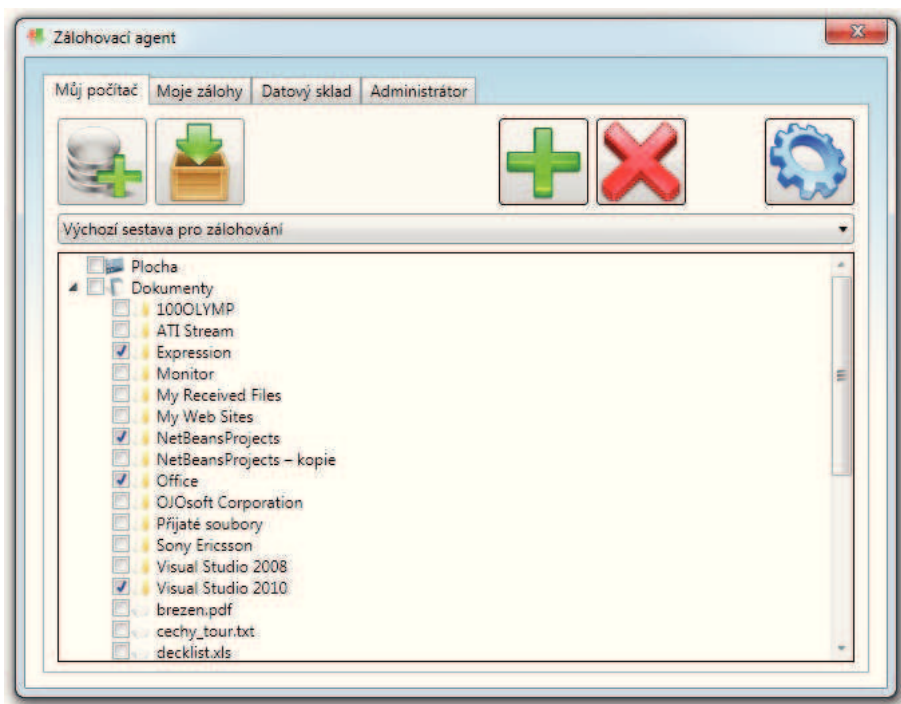
6.1.4 Grafické uživatelské rozhraní

Grafické uživatelské rozhraní (GUI) je část aplikace, která prostřednictvím interaktivních ovládacích prvků umožňuje uživateli danou aplikaci ovládat.

Čtyřmi nejdůležitějšími částmi GUI této aplikace jsou hlavní okno aplikace, notificační ikona v oznamovací oblasti s vysouvacím menu základní nabídky programu, okno

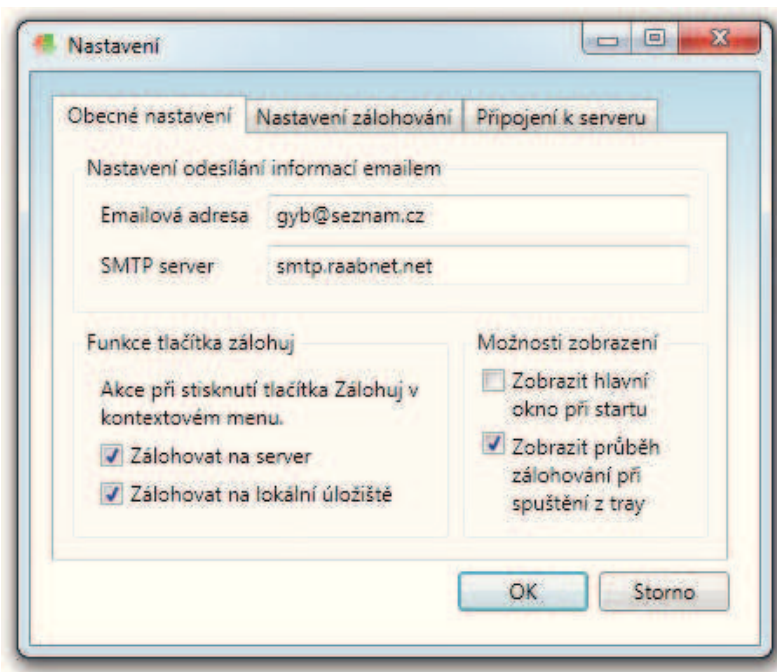
nastavení a informační dialog průběhu procesu zálohování či obnovy. GUI aplikace dále obsahuje několik dalších oken a dialogů vztahujících se vždy k určité volbě či nastavení.

Hlavní okno umožňuje uživateli samotné spuštění operací zálohování, archivace a obnovy, a to pomocí tlačítek v horní části vybrané záložky. Záložka Můj počítač slouží ke specifikaci souborů a adresářů určených k zálohování. Jednotlivé sestavy lze ukládat jako předvolby pro opětovné použití, což značně zjednodušuje práci s programem. Záložka Moje zálohy slouží k prohlížení již vytvořených záloh, jejich správě a k případné obnově jednotlivých souborů a složek či celé zálohované sestavy. Záložka datový sklad (zobrazí se pouze při připojení k centrálnímu serveru) slouží k prohlížení datového archivu společnosti. Tato činnost může být omezena přidělenými uživatelskými právy konkrétního uživatele. Záložka Administrátor (zobrazí se pouze při připojení k serveru administrátorským účtem) umožňuje správci vzdálené nastavení serveru, zobrazení jeho stavu a správu uživatelů.



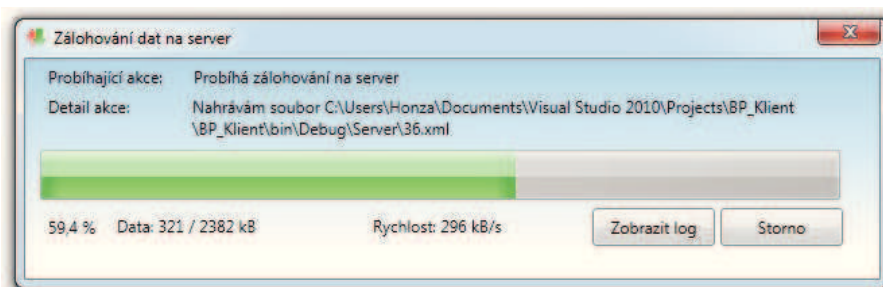
Obrázek 16: Hlavní okno grafického uživatelského rozhraní

Okno Nastavení slouží uživateli k základnímu nastavení aplikace. Ve třech záložkách přehledně sdružuje všechny volby důležité pro správný běh celého programu. Má podobný design jako hlavní okno, což umožňuje uživateli snadnou orientaci.



Obrázek 17: Okno nastavení aplikace

Informační okno poskytuje uživateli detailní informace o průběhu zpracovávané operace. Lze z něj vyčíst aktuálně zpracovávanou akci, její popis, celkový stav průběhu, rychlost a kvantitu přenášených dat. Po kliknutí na tlačítko Zobrazit log může uživatel nahlédnout do podrobného reportu o průběhu zpracování operace.



Obrázek 18: Informační okno aplikace

6.2 Implementace serveru

Implementace databázové části systému vychází z datové analýzy. Po vytvoření všech databázových tabulek a definování procedur a funkcí byly změřeny všechny dotazy a aktualizace. Z těchto měření byla poté navržena vhodná indexace atributů jednotlivých tabulek tak, aby byl maximálně navýšen výkon celého SŘBD. Většina funkcí systému, co manipulují s daty v databázi, jsou z důvodů lepšího oddělení obou částí architektury klient-server prováděny uvnitř SŘBD.

7 Závěr

7.1 Nasazení systému v praxi

První fáze testování, ladění a ověřování funkčnosti je úzce spjata se samotným procesem implementace. Během této fáze byly zjištěny a opraveny všechny chyby systému, které ovlivňují jeho správnou funkčnost.

Pro ověření, že systém plně vyhovuje požadavkům zadavatele, byl tento nainstalován do virtuálního prostředí a nasazen do testovacího provozu. V této fázi testování při omezeném počtu uživatelů nebyly prozatím odhaleny žádné vážné nedostatky ve funkcionalitě a běhu systému. Ten však bude dlouhodoběji testován (v řádu několika týdnů až měsíců), než bude plně odladěn a zařazen do plného provozu.

Pro správnou funkci klientské aplikace je nutné mít nainstalovaný .NET Framework 4 a Oracle Data Access Components (ODAC), jež jsou součástí přílohy této bakalářské práce, a to v 32-bitové i 64-bitové verzi.

7.2 Možná rozšíření systému

Navržený a implementovaný systém centralizovaného úložiště dat realizuje přesně požadavky zadavatele dle jejich analýzy, a to v rámci rozsahu této bakalářské práce. Po konzultaci se zadavatelem bylo rozhodnuto pokračovat ve vývoji tohoto systému i do budoucna. Možných rozšíření aplikace se nabízí celá řada:

- lepší rozhraní administrátora
- duplikace úložišť dat
- komprese dat na serveru
- zavedení více schémat zálohování
- deduplikace na úrovni binárních dat aj.

Systém byl navržen tak, aby bylo možné jej dále rozšiřovat o novou funkcionalitu. Postup dalšího vývoje bude závislý na budoucích požadavcích zadavatele, výsledcích dlouhodobého nasazení a na zpětné vazbě od uživatelů tohoto systému.

7.3 Závěrečné zhodnocení

Cílem této bakalářské práce bylo seznámit se s problematikou zálohování a archivace dat v prostředí malých až středních firem, seznámit se s existujícími nástroji postihujícími danou problematiku, analyzovat požadavky zadavatele, společnosti Vodovody a kanalizace Přerov, a.s. a navrhnout a prakticky realizovat aplikaci řešící zmíněnou agendu. Výsledkem této práce je tedy systém zálohování a archivace dat implementovaný na platformě Microsoft .NET.

Zadaná problematika je natolik obsáhlá, že není v mezích této bakalářské práce obsáhnout všechny oblasti s touto problematikou spojené. Mnou navržený a realizovaný systém však může posloužit jako základ pro vývoj více sofistikovaného řešení.

Literatura

- [1] Jana Šarmanová, Informační systémy a datové sklady, učební text, Ostrava 2007.
- [2] Amadeo Mareš, 1001 Tipů a triků pro C#, Computer Press, Brno 2008.
- [3] Chris Zeis, Chris Ruel, Michael Wessler, Oracle 11g for Dummies, Wiley Publishing Inc., Indianapolis 2009.
- [4] <http://www.wikipedia.org>, Wikipedie, otevřená encyklopedie.
- [5] <http://www.storage.cz>, Data Storage technologie.
- [6] <http://www.zalohovani.info>, Hewlett Packard partner.
- [7] <http://education.oracle.com>, Oracle Academy.
- [8] <http://msdn.microsoft.com>, Microsoft MSDN Library.

Přílohy

[1] DVD disk obsahující text bakalářské práce v elektronické podobě a zdrojové kódy